

zhao_zi_ming_lc的博客

RSS订阅

初级安卓开发工程师、python爱好者的成长之路

python Web开发 flask轻量级Web框架

2017年11月10日 18:19:18

阅读数：1564



0



写评论



收藏



微信



微博



QQ

Q、flask介绍

Flask是一个使用 Python 编写的轻量级 Web 应用框架。其 WSGI 工具箱采用 Werkzeug，模板引擎则使用 Jinja2。Flask使用 BSD 授权。Flask也被称为“microframework”，因为它使用简单的核心，用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。

Web应用采用访问和响应的方式和用户进行交互。

主要由用户发起一个Web request 请求，对网页进行请求。

请求成功后由Web 服务器放回一个 response 对象对浏览器进行渲染，显示出返回的网页。

浏览器访问网页的默认是get方法

flask应用的默认ip是本地IP 127.0.0.1 默认端口是5000

一、flask应用的基本构成

```
1 #coding=utf-8
2 from flask import Flask
3 app = Flask(__name__)
4
5 @app.route('/')
6 def hello_world():
7     return "hello world"
8
9 if __name__ == '__main__':
10     app.run()
```

1.

`app = Flask(name)`

这是建立一个flask应用，并且设置flask应用的目录为该模块的目录。

2.

`@app.route('/')`

设置应用的路由，既访问的Web服务器的目录。

3.

`@app.route('/')``def hello_world():``return "hello world"`

设置路由函数，访问该路由时执行该函数。

4.

`app.run()`

启动flask应用

运行程序，在浏览器中访问服务器地址即可得到返回的网页

二、flask路由和反向路由

1.参数的传递

在应用中添加一下代码，在访问WebUrl中添加参数可以把参数传递给Web后台程序。

注意：传进去的值是以字符串形式存在。

```
1 @app.route('/users/<id>')
2 def hello_world(id):
```

登录

注册

X

← → ↺ ↻ 127.0.0.1:5000/users/55

users: 55

http://blog.csdn.net/zhao_zi_ming_lc

2 @route路由中的访问方法

路由默认的是get方法，也可以更改路由方法，例如

```
1
2 @app.route('/user', methods=['POST'])
3 def hello_user():
4     return "hello user"
```



0



3.通过request 库来传递id

得添加requests库的导入



收藏



微信



微博



QQ

```
1 from flask import Flask,request
```

```
1 @app.route('/query_user')
2 def query_user():
3     id = request.args.get('id')
4     return "query_user: id " + id
```

属性输入采用 url + ?属性名=属性值的格式

← → ↻ 🏠 ⓘ 127.0.0.1:5000/query_user?id=123456

query_user: id 123456

4.反向路由

通过视图函数反导出url地址

```
1 @app.route('/query_url')
2 def query_url():
3     return 'query_url' + url_for('query_user')
```

这样就在query_url函数中访问到了 query_user的路由地址

← → ↻ 🏠 ⓘ 127.0.0.1:5000/query_url

query_url/query_user

二、flask模板

模板是为了使后台逻辑代码与网页前端代码分离。

先引入render_template模块

from flask import render_template

新建一个template文件夹，存放html文件。

1.模板的基本使用

例如demo.html文件

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
```

登录

注册



```
9 </body>
10 </html>
```

```
1 @app.route('/')
2 def hello_world():
3     return render_template("demo.html", content="hello flask ")
```

← → ↻ 🏠 ⓘ 127.0.0.1:5000

hello flask hello world this is first web



0

写评论

收藏

微信

微博

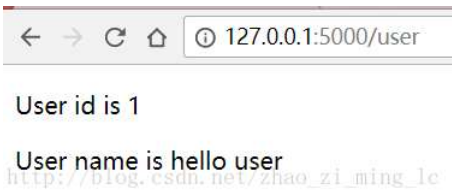
QQ

2.传入类

```
1
2 class User:
3     def __init__(self, user_id, user_name):
4         self.user_id = user_id
5         self.user_name = user_name

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8 <p>User id is {{ user.user_id }}</p>
9 <p>User name is {{ user.user_name }}</p>
10 </body>
11 </html>

1 @app.route('/user')
2 def hello_user():
3     user = User(1, "hello user")
4     return render_template("demo.html", user=user)
```



1. 模板的 if 语法

query_user.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8 {% if user %}
9 {{ user.user_name }}
10 {% else %}

14 </body>
    </html>
```

登录 注册 ×

如果user不为空则输出 user.name 否则 输出hello world

```
1 @app.route('/query_user/<id>')
2 def query_user(id):
3     user = None
4     if id == '1':
5         user = User(2, "zzm")
6     return render_template("query_user.html", user = user)
```

0

写评论

收藏

微信

127.0.0.1:5000/query_user/1

zzmhttp://blog.csdn.net/zhao_zi_ming_lc

127.0.0.1:5000/query_user/3

hello worldhttp://blog.csdn.net/zhao_zi_ming_lc



微博



QQ

4.模板循环

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8 {% for user in users %}
9     {{ user.user_id }}----{{ user.user_name }}<br/>
10 {% endfor %}
11 </body>
12 </html>

1 @app.route('/list')
2 def user_list():
3     users = []
4     for i in range(1, 11):
5         user = User(1, "zzm" + str(i))
6         users.append(user)
7     return render_template("user_list.html", users=users)

```

← → ↻ 🏠 ⓘ 127.0.0.1:5000/list

```

1----zzm1
1----zzm2
1----zzm3
1----zzm4
1----zzm5
1----zzm6
1----zzm7
1----zzm8
1----zzm9
1----zzm10

```

[登录](#)[注册](#)[×](#)

5.模板继承

以下代码省略HTML主体结构，只下body中的部分
先写一个base.html

```

1 <div>
2     这是头部
3 </div>
4 {% block content %}
5 {% endblock %}
6 <div>
7     这是尾部
8 </div>

```



0

然后在one_base.html中用extends 语法继承base.html页面，
然后再重写content中的内容



写评论



收藏



微信



微博



QQ

运行结果

← → ↻ 🏠 ⓘ 127.0.0.1:5000/one

这是头部

这是第一页

这是尾部

三、消息提醒和异常处理

1.消息提醒

在网页中设置提醒的消息需要引入flash模块

from flask import flash

flash.html

```
1 <form action="/login" method="post">
2     <input type="text" name="username">
3     <input type="password" name="password">
4     <input type="submit" value="提交">
5 </form>
6 <h2>{{ get_flashed_messages()[0] }}</h2>
```

```
1 @app.route('/flashDemo')
2 def flash_demo():
3     flash("hello world flash")
4     return render_template("flash.html")
```

← → ↺ ↻

127.0.0.1:5000/flashDemo

提交

hello world flash

<http://blog.csdn.net/zhao zi ming lc>

登录 注册 ×

2.异常处理

在别人输错网址的时候，服务器会返回一个404异常，将会返回一个浏览器的404网页。

我们可以进行异常处理，抛出自己写的404网页，这样用户体验比较好。

写个404.html

```
1 <h1>this is error</h1>
```

异常处理语法：

errorhandler() 括号里面填写错误代码

```
1 @app.errorhandler(404)
2 def not_fond(e):
3     return render_template("404.html")
```

这里随便输错一个网页，就弹出来了我们自己编写的404异常网页

← → ↺ ↻

127.0.0.1:5000/bc

this is error



0



写评论



收藏



微信



微博



QQ

喜欢python的小伙伴可以关注下我的公众号
纯干货技术文章分享

